

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR PATENT

ON

**METHOD FOR DETERMINING SOFTWARE LOAD PROCESS**

BY

KEITH C. THOMAS  
624 STERLING  
VERMILLION, SD 57069  
CITIZEN OF USA

## **METHOD FOR DETERMINING SOFTWARE LOAD PROCESS**

### **FIELD OF THE INVENTION**

[0001] The present invention generally relates to digital computers, and particularly to a method for determining the software load process.

### **BACKGROUND OF THE INVENTION**

[0002] Almost every computer user has experienced such a situation: s/he is working on something on the computer and suddenly the computer either stops responding (“freezes”) or shows the “blue screen of death” (“crashes”). There are several indications that let a user know that a computer may be frozen. The most common is the “unresponsive” program. The program will simply not do what you tell it to do. Additionally, the program may appear “stuck” in the task bar at the bottom of the screen (it will appear to be eternally minimized, never being capable of being opened for use by a user). Alternatively, a user may see a message pop up, telling the user that a program has performed an “illegal operation” or “is no longer responding;” these are ways the computer operating system tells the user that the program has frozen. A major indication that a computer may be frozen is that the mouse “freezes” in its place, and cannot be moved about the screen any more. The “blue screen of death” is probably the most obvious indication: a computer will flash to a blue-background with white text, listing a small amount of computer code alongside an error message.

[0003] There are many reasons why a computer may freeze or completely crash, among which the system software load process may be a major reason. Those of ordinary skill in the art will understand that software is a generic term for organized collections of computer data and instructions. Software is often broken into two major categories: system software that provides the basic non-task-specific functions of the computer, and application software which is used by users to accomplish specific tasks.

[0004] The computer system software load process may be critical to understanding and resolving the issue why a computer freezes or crashes. Thus, it would be advantageous to provide a method to determine the different system software load process: whether the loaded computer system software was loaded in a computer manufacturer's factory, loaded by a system restoring CD (SRCD), or loaded by downloading an image. By communicating the system software load information back to a central authority when or after a computer crashes or freezes, the computer manufacturer may be able to use such information (e.g., by performing a statistical analysis of the data, and the like) to trace the cause of why the computer crashes or freezes and find a corresponding solution.

#### SUMMARY OF THE INVENTION

[0005] Accordingly, the present invention is directed to a method for determining the software load process. In an exemplary aspect of the present invention, a method for determining computer system software load process may start with loading computer system software on a computer. Then a marker file may be provided in the loaded computer system software. The marker file may indicate whether the loaded computer system software was loaded in a computer manufacturer's factory, loaded by a system restoration CD (SRCD), or loaded by downloading an image. When a program running on the computer abnormally terminates and the computer is able to send the marker file to a central authority, the marker file may be sent to the central authority by the computer. When a program running on the computer abnormally terminates and the computer is not able to send the marker file to a central authority, the marker file may be sent to the central authority by the computer after the computer reboots.

[0006] It is understood that the present invention may be applied to determining a load process for any computer software, including any system software and any application software.

[0007] It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention as claimed. The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate an embodiment of the invention and together with the general description, serve to explain the principles of the invention.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The numerous advantages of the present invention may be better understood by those skilled in the art by reference to the accompanying figures in which:

FIG. 1 shows an exemplary method for determining computer system software load process in accordance with an exemplary embodiment of the present invention; and

FIG. 2 shows an exemplary method for determining computer software load process in accordance with an exemplary embodiment of the present invention.

#### DETAILED DESCRIPTION OF THE INVENTION

[0009] Reference will now be made in detail to the presently preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings.

[0010] Referring first to FIG. 1, an exemplary method 100 for determining computer system software load process in accordance with an exemplary embodiment of the present invention is shown. The method 100 may start with step 102, in which computer system software is loaded onto a computer 102. Those of ordinary skill in the art will understand that system software is a generic term referring to any computer program or library whose purpose is to help run the computer system, as opposed to application software that helps solve user problems directly. System software includes operating systems (e.g., a Windows operating system, LINUX, and the like), device drivers, compilers, assemblers, linkers, utilities, and the like.

[0011] Next, a marker file may be provided in the loaded computer system software, the marker file indicating whether the loaded computer system software was loaded in a

computer manufacturer's factory, loaded by a system restoring CD (SRCD), or loaded by downloading an image (e.g., from the Internet, an intranet, the World Wide Web, and the like) 104. In a preferred embodiment, when the loaded computer system software is loaded by downloading an image, the marker file may be dynamically generated. In an additional embodiment, when the loaded computer system software is loaded by a SRCD, the marker file in the loaded computer system software may be a static marker file loaded from the SRCD. In another embodiment, when the loaded computer system software is loaded in a computer manufacturer's factory, the marker file may be either dynamically generated or may be a static marker file loaded by the factory. It is understood by those of ordinary skill in the art that the marker file may be generated in various embodiments without departing from the scope and spirit of the present invention. The image, the system software, the marker file, or components thereof, may be loaded by any of the manufacturer, the end user, or a maintenance representative. The manufacturer may be the original design manufacturer ("ODM"—also known as product supplier; e.g., a manufacturing concern under contract to the OEM such as Wistron), the third party installer, the original equipment manufacturer ("OEM"; e.g., Gateway), or the like.

[0012] Then, when a program running on the computer abnormally terminates (e.g., when the computer "freezes" or "crashes" or the like) and when the computer is able to send the marker file to a central authority, the marker file may be sent to the central authority by the computer 106. Next, when a program running on the computer abnormally terminates and when the computer is not able to send the marker file to a central authority, the marker file may be sent to the central authority by the computer after the computer reboots 108. A central authority may be a computer manufacturer, a computer system software developer, and the like.

[0013] Referring to FIG. 2, an exemplary method 200 for determining computer software load process in accordance with an exemplary embodiment of the present invention is shown. The method 200 is similar to the method 100 shown in FIG. 1. However, the method 200 may be applied to determining a load process for any software,

including system software and application software. The method 200 may start with a step 202, in which computer software is loaded onto a computer 202. A marker file may be provided in the loaded computer software, the marker file indicating whether the loaded computer software was loaded in a computer manufacturer's factory, loaded by a computer-readable medium (e.g., conventional floppy disks, optical disks, CD-ROMs, magneto-optical disks, or the like), or loaded by downloading an image (e.g., from the Internet, an intranet, the World Wide Web, and the like) 204. In a preferred embodiment, when the loaded computer software is loaded by downloading an image, the marker file may be dynamically generated. In an additional embodiment, when the loaded computer software is loaded by a SRCD, the marker file in the loaded computer software may be a static marker file loaded from the SRCD. In another embodiment, when the loaded computer software (e.g., MS Word, and the like) is loaded in a computer manufacturer's factory, the marker file may be either dynamically generated or may be a static marker file loaded by the factory. It is understood by those of ordinary skill in the art that the marker file may be generated in various embodiments without departing from the scope and spirit of the present invention.

[0014] Then, when a program running on the computer abnormally terminates (e.g., when the computer "freezes" or "crashes" or the like) and when the computer is able to send the marker file to a central authority, the marker file may be sent to the central authority by the computer 206. Next, when a program running on the computer abnormally terminates and when the computer is not able to send the marker file to a central authority, the marker file may be sent to the central authority by the computer after the computer reboots 208. A central authority may be a computer manufacturer, a computer system software developer, and the like.

[0015] The present invention may provide a method for determining the type of download that was on a computer system before it "freezes" or "crashes" and may provide error reporting data to a central authority in order to assist computer manufacturers in tracking computer issues. By communicating the system software load

information back to a central authority when or after a computer crashes or freezes, the computer manufacturer may be able to use such information (e.g., by performing a statistical analysis of the data, and the like) to trace the cause of why the computer crashes or freezes and find a corresponding solution.

**[0016]** It is to be noted that the above described embodiments according to the present invention may be conveniently implemented using conventional general purpose digital computers programmed according to the teachings of the present specification, as will be apparent to those skilled in the computer art. Appropriate software coding may readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art.

**[0017]** It is to be understood that the present invention may be conveniently implemented in forms of software package. Such a software package may be a computer program product which employs a storage medium including stored computer code which is used to program a computer to perform the disclosed function and process of the present invention. The storage medium may include, but is not limited to, any type of conventional floppy disks, optical disks, CD-ROMs, magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, or any other suitable media for storing electronic instructions.

**[0018]** It is understood that the specific order or hierarchy of steps in the processes disclosed is an example of exemplary approaches. Based upon design preferences, it is understood that the specific order or hierarchy of steps in the processes may be rearranged while remaining within the scope of the present invention. The accompanying method claims present elements of the various steps in a sample order, and are not meant to be limited to the specific order or hierarchy presented.

**[0019]** It is believed that the present invention and many of its attendant advantages will be understood by the foregoing description. It is also believed that it will be apparent that various changes may be made in the form, construction and arrangement of the

components thereof without departing from the scope and spirit of the invention or without sacrificing all of its material advantages. The form herein before described being merely an explanatory embodiment thereof, it is the intention of the following claims to encompass and include such changes.